

Задача ПОДГОТОВКА

Подзадача 1

Обхождаме всички числа и броим колко са четни. Ако ни мързи просто извеждаме $n/2$.

Подзадача 2 Може да намерим цифрите на числото и да видим дали техният брой е 3. Ако ни мързи проверяваме дали числото е в интервала $[100; 999]$.

Подзадача 3 и 4 Тук вече трябва да обходим цифрите на числото. Гледаме последната цифри и я махаме. Повтаряме това докато има цифри. Намирането на сумата цифри и броя петици за едно число n би изглеждал така:

```
int sum = 0;
int cnt = 0;
while (n != 0) {
    // разглеждаме цифрата n%10
    sum += n%10;
    if (n%10 == 5) cnt++;
    n/=10;
}
```

Подзадача 5 Търсим броя числа, които са степени на 2. Може да проверим дали това е изпълнено за всяко число поотделно. За произволно число ще го делим на 2 докато можем. Ако накрая остане 1, то значи е степен на 2.

```
bool isPow2(int n) {
    while (n%2 == 0) n/=2;
    return n == 1;
}
```

Подзадача 6

По подобие на предната подзадача може да делим последователно на 1, 2, 3, ... докато можем. Ако в края остане 1 то условието е изпълнено.

Подзадача 7 За да проверим дали едно число е палиндром трябва да обърнем цифрите му. Стандартно като обхождаме цифрите на едно число започваме с последната, после предпоследната и чак накрая първата. На обърнатото

число първо намираме първата цифра от ляво надясно, после втората и т.н. Трябва да знаем как да добавим цифра в края на число. Основната част от задачата е ако имаме числото n и цифрата d как да добавим d в края на n . Решението е на две стъпки - първо добавяме 0 в края на n , което става с умножение по 10, след което към резултата добавяме d . Ето как изглежда да обърнем числото n :

```
int opposite(int n) {
    int m = 0;
    while (n != 0) {
        m = 10*m+n%10;
        n /= 10;
    }
    return m;
}
```

Подзадача 8 В тази задача проверката дали едно число е част от редицата е по-сложна. По-лесния подход е да започнем да генерираме числата от редицата и да спрем в момента, в който получим число, което е по-голямо от n .

Петър Петров